Garabato: A proposal of a Sketch-Based Image Retrieval System for the Web

Ana María Miguelena Bada Guillermo de Jesús Hoyos Rivera Antonio Marín Hernández

Abstract— **A proposal for a queried-by-sketch image retrieval system is introduced as an alternative to text-based image search on the Web. The user will create a sketch as a query that will be matched with the edges extracted from natural images. The main challenge regarding edge detection for Content-based Image Retrieval consists in finding edges for larger regions and avoiding the ones corresponding to textures. For this purpose, a combination of selective smoothing and color segmentation is applied prior edge extraction. An evolutionary algorithm is deployed to optimize the image-processing parameters. Similarity between the user´s sketch and the image's edges will be measured regarding two local aspects: spatial proximity and edge orientation. A full architecture for image search on the Web is proposed and preliminary results are reported using a trial database.**

I. INTRODUCTION

Our project, named Garabato, arises as an effort to enable the capability of communication via sketches with the computer, with the specific task of retrieving images. This can be helpful for searching on a database with images that are not labeled so that text-based search is not possible. However, this project is primarily oriented to image search on the Web because, nowadays, is the greatest and most used source of images.

Even though typical Web search engines have the images indexed by text labels, the results of the retrieval may depend on the language of the labels and the accuracy of the correspondence between text and visual content. This query by sketch proposal brings up the possibility to find specific shapes and objects displayed on images without concerning the text surrounding or describing it on the webpage.

Currently most used image search engines are queried by a string of characters, although this is an incomplete approach since characters and multimedia content cannot be directly corresponded. Content-based Image Retrieval (CBIR) systems have arisen as alternatives to text-based image search, attempting to bridge the gap between visual and semantic-free text while retrieving images from the Internet or a particular database. Since 1995, several systems have been proposed and developed, varying the type of query, the features and the descriptors of the images and the specific database or application. Some comprehensive surveys of CBIR history, trends and perspectives can be found in [1] and [2]. However, many proposals never transcended the prototype version. There are some state of the art CBIR systems available to search on the Web, but they all are queried by image example and not by sketch.

Authors are with the Department of Artificial Intelligence, Universidad Veracruzana, Sebastián Camacho No. 5, Xalapa, Ver., 91000, Mexico. [annamiguelena, ghoyosr] at gmail.com, anmarin at uv.mx

Garabato is a Content-based Image Retrieval System (CBIR) queried by sketch; this means that the user can create the visual content to match the images on the Web. This alternative is very interactive for the user, and can be used to retrieve images regardless the language of the Web pages where they are found.

The sketches must be hand-made with no text, context around objects or black areas, following the specifications described in [3]. To match the sketches, images need to be pre-processed in order to extract the features that can be used to evaluate similarity with a sketch.

The proposal addresses two basic problems: *i)* contours or salient edges must be extracted from each image of the database in order to present similar characteristics to a sketch; *ii)* sketch and edge images must be analyzed to represent and quantify their characteristics, in other to compute the similarity between them.

In the following section, related work to CBIR systems for the Web is presented as well as some background on salient edge detection and similarity measuring for edge representations. In section III the proposal architecture of Garabato is displayed. An evolutionary approach is presented to find the salient edges of images and a similarity measure for edges is described in sections IV and V, respectively. In section VI preliminary results with a trial database are reported. Finally, conclusions are presented along with future work prospects.

II. RELATED WORK

The most famous image search engine on the Web, Google, has added a new modality called *Search by Image* [4], where users can upload an image from a local source or provide an image's URL. As a result, the engine return a series of pages containing the exact same image or some that are considered to be similar to it. It also can be asked to return only visually similar images. Yet, this similarity relies in comparing colors and textures and the spatial distribution of these two, aided by textual content from the pages where these are located. This approach can be very useful when the visual example can be identically found on the Web, otherwise, the engine may fail to find images containing the same type of objects. This kind of search is often called *reverse images search* and there are several projects providing this service on the Web: TinEye [5], Revimg [6] and Wesee [7].

Some approaches for sketch-based image retrieval are based on color sketches, like Retrievr [8], in which the result

978-1-4799-3469-0/14/$31.00 ©2014 IEEE

images are selected according to the color distribution described on the sketch. Related work on edge-like sketches for Content-based Image Retrieval is presented concerning two essential elements: salient edge extraction and similarity calculation methods.

In [9], the authors proposed to add a step called Surround Suppression to the Canny edge extraction algorithm; this aims at suppressing texture edges and improving detection of objects contours and region boundaries in natural scenes. The step consists in adding an inhibition term for every point of the image, which depends on the values of the gradient on the surrounding of the concerning point. Edges form textures must present higher inhibition values and, consequently, they will be ignored while computing Canny edge detection.

Local image measurements are used in [10] to detect natural image boundaries. A classifier is trained using human labeled images to obtain the probability that a given pixel is a boundary, depending of local values as brightness and color.

A morphological operator called Adaptive Pseudo Dilatation (APD) is introduced in [11], which uses context dependent structuring elements in order to identify long curvilinear structures. The group of connected edges obtained is shown to be consistent with the Gestal law of good continuation. The ADP is applied several times with different thresholds. Each time a group of edges is removed and contours are completed.

Some methods that use a saliency map to identify where is the most attention-driven object in the image and then use special techniques to extract the contours are shown in [12] and [13]. For a survey in contour detection [14] can be consulted.

Regarding edge description and similarity, in [1] and [15] a bag-of-features descriptor is used. It consists in calculating local features like shape context [16], histogram of oriented gradient [17] and two more features proposed by the author. These are computed for each section of the windowed images, as for the sketch query. The system searches for the images that have the most similar feature values with the sketch in the correspondent window.

Chamfer Matching is proposed in [18] as a pyramidal algorithm that matches edges by minimizing a generalized distance between them. Recently this edge matching has been re-implemented as part of the MindFinder project [19] where the images of edges are divided in channels for each edge orientation, and then, Chamfer matching is applied separately on each channel.

III. Proposal for a Sketch-Based Image Retrieval System

This Image Retrieval system that we propose has a structure as displayed in Fig. 1. As it can be clearly seen, there are two main working blocks. On the side, there is an offline process, in which the images are collected from the Web, and then pre-processed so only the features needed to compare with a sketch are saved in the database. On the other hand, there is an online process, where the user sets its query as a sketch, which is matched with the elements in the database, so he receives as results the images which features made the best match.

To collect images from the Internet, there will be an independent program continuously browsing Web pages and locating images. Each image will be passed as an input to the Feature Extraction Module. The original images will not be stored in the database, only the feature descriptors linked to the URL of the original image.

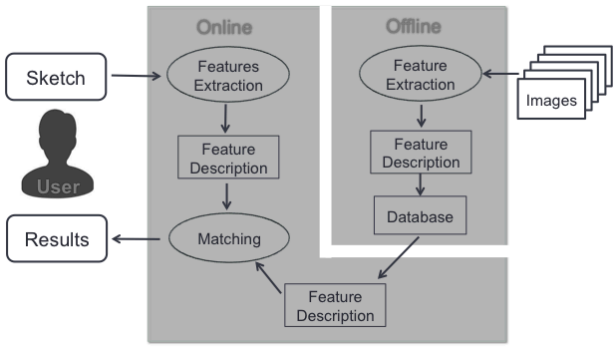


Figure 1. Architecture of the proposed image retrieval system.

The Feature Extraction module is in charge of processing the rough images as found on the Web, extracting features that can be compared with the query. In this case, the features are set to be Salient Edges, corresponding to the boundaries of the coarser regions. The output after the edge extraction is a binary image of the edges. This image is then transformed to a more descriptive representation that can be directly used to compare it with the sketch. These representations are called Feature Descriptors; for this proposal, the descriptors are orientation maps, which can be stored as compact files in the system's database, but keeping a relation with the original image's URL. Another main module of the Garabato's architecture is the one in charge of Matching the descriptors of the user's sketch with the ones saved on the database, establishing a similitude measure for each of the descriptors analyzed.

To make better matches between the sketches and the image's features, it is necessary to extract the edges of the image, but only the ones that represent objects and larger regions. These are usually called Salient Edges or Contours. Every algorithm for simple edge extraction will typically fail to obtain the salient edges for natural images, because they are based upon the detection of intensity variation, yet in most cases, some important boundary edges may present less contrast than some texture-related edges. To aid this difficulty, we propose to extract the salient edges of images by finding regions with similar colors and textures, then applying the Canny algorithm to obtain connected and thin edges.

The presented approach extracts salient edges by a combination well-known methods for bilateral smoothing, segmenting by color and an edge extractor. The asset of our proposal is that the parameters for each image-processing algorithm are generated using a differential evolution algorithm that uses local characteristics to evaluate how well a configuration of parameters perform for an image. In this way, the method adapts to each image allowing to compound a database with diverse images.

There are two basic approaches to measure similarity between a sketch and a set of images: by matching the shape or

by matching the edges. Shape matching has the advantage of being invariant over affine transformations but is restrictive by requiring the sketch to be a closed contour. In this project shape matching will not be considered so the user will have more freedom to draw the sketch.

The descriptors that we propose for the edges are based on the orientation of each edge pixel. Differing from other approaches, the orientation is not calculated using the gradient, but by analyzing the structure of the pixels on the neighborhood using our proposed line features. In this way the orientation is not calculated depending of the gray level intensities, which are not available for the user's sketch.

IV. Feature Extraction: Salient Edges

Canny edge detector [20] uses a Sobel kernel to detect the borders and then applies post processing for: *i)* generating thin edges, so the final output will be composed by one-pixel-width curves; *ii)* selecting edges according to the gradient magnitude, applying a non-maximal suppression that depends on two thresholds; *iii)* closing opened edges by completing with nearby edges that follow on the same direction. The main advantage of Canny over other edge detection methods is that it generates uniform edges with the same width, most edges are connected and a lot of noise edges are filtered out. However, the output depends greatly on the three parameters: the Sobel kernel size k_s, and the two thresholds, T_1 and T_2, for the non-maximal suppression.

To reduce the contrast within textures but not objects boundaries, a Gaussian bilateral filtering [21] is applied. This type of smoothing operates in the spatial and intensity domain working as an edge-preserving smoothing but, if edges are close together within the neighborhood, they are smoothed. If the smoothing is applied on pixel *x*, the pixel's neighborhood is denoted by *N(x)* , an so the smoothed pixel is obtained by:

$$H(x) = \frac{1}{C} \sum_{y \in N(x)} e^{\frac{-\|y-x\|^2}{2\sigma_d^2}} e^{\frac{-|I(y)-I(x)|^2}{2\sigma_r^2}} \quad (1)$$

As with the edge detector, the output depends on the parameters σ_d and σ_r, which are the standard deviation for the Gaussian kernel in the space and intensity domain, respectively. To obtain an improved effect of texture reduction the smoothing can be applied several times. Among with σ_d and σ_r, the neighborhood size N_s and the amount of times that it is applied, n, are also considered parameters conditioning the output.

After the bilateral smoothing, the images are segmented by color and spatial proximity using a pyramidal mean-shift algorithm [22]. This algorithm receives as parameters two neighborhood radius, s_p and s_r, which are radius for spatial and color neighborhoods; these are used to define a window for each pixel. Iterations start with every pixel's window centered on it, for each window the mean is computed. Then it shifts the center of the window to the mean and repeats the algorithm until it converges. Pixels, for which the starter windows converge to the same point, are merged into regions. The pyramidal approach starts with a lower resolution of the image and inherits the result to the next resolution where the process is repeated. The number of the levels of the pyramid, L, is also a parameter for the algorithm.

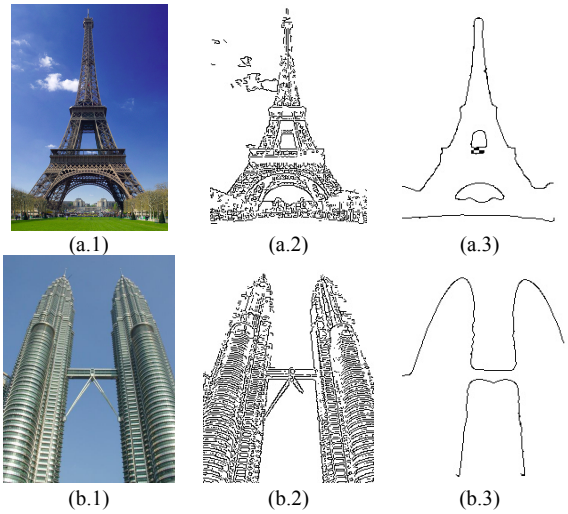


Figure 2. Sample Canny edge extraction (column a) and salient edge extraction (column b).

As shown in Fig. 2b, the combination of bilateral filtering, mean-shift segmentation and Canny Edge Detector can lead to the extraction of salient edges. Although further experimentation with a larger set of images supported that this is a valid approach for salient edge detection, it also revealed the difficulty to establish the right combination of the image-possessing parameters to assure a satisfactory result for every image. To solve this inconvenient, an evolutionary approach is proposed to find, for every image, a configuration of the image-possessing parameters.

The differential evolution algorithm [23] iteratively searches the best candidate solution according to a fitness function. This method makes no assumptions about the problem and can search between large or short ranges of the solution candidates. It does not require the gradient of the candidate solution, so it can also be used to optimize non-continuous values. The set of parameters is defined as follows

$$\{n, N_S, \sigma_d, \sigma_r, s_p, s_r, L, k_s, T_1, T_2\}, \quad (2)$$

where each coordinate has the interpretation as in the description of the image processes. All of them are positive integer values and the range of numbers that each can take depends on the definition for each parameter. For example, N_S and k_s are convolution kernel sizes, which must be odd positives values in the units scale, so they are optimized between the values 3, 5, 7 and 9; in other hand T_1 and T_2 are thresholds for grey levels, so they can be between 0 to 255.

One main feature of evolutionary algorithm is the fitness function that should be able to evaluate the goodness of a solution according to the characteristics of the problem. In this case, the solution vector cannot be evaluated directly from its values. It is necessary to evaluate how well they perform in the salient edge extraction process for a particular image. For this purpose, each candidate solution must be used as the parameters of the bilateral smoothing, mean-shift segmentation and canny edge detection algorithm, applied on a particular

image. The resulting image must be evaluated in order to determine how well it fits as the solution of the problem.

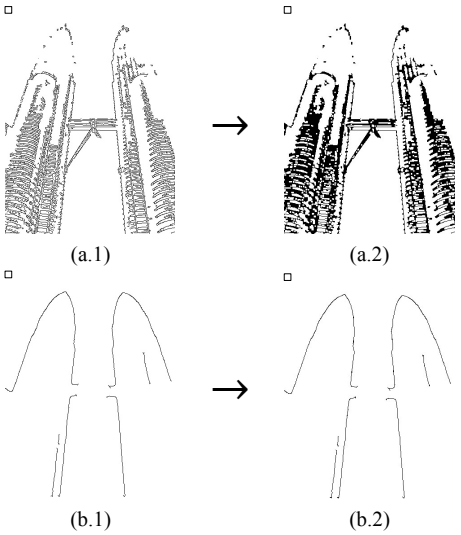
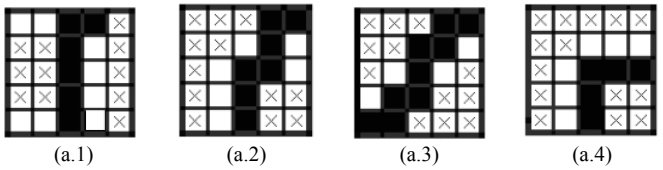


Figure 3. The effect of morphological closing (images are displayed with inverted colors to improve visibility, but closing was applied with black background and white edges). The image a.1 present a great amout of edges corresponding to textures. After closing, in a.2 those edges were joint together in black regions. In an image forme only by smooth curves, there is not mayor effect after closing (row b.).

To define the fitness function, the following observation is made: edges detected due to fine structures and textures will be very close. As edges are generated from Canny edge detection, all curves are 1-pixel width. Following this, if a morphological close operation is applied the edges that were close together will form black areas, while isolated ones will remain the same (Fig. 3). Then, to quantize the goodness of the solution it is necessary to count the amount of pixels that remain as thin lines after the morphological close.



0	0	1	1	-2
-2	-2	1	0	-2
-2	-2	1	0	-2
-2	-2	1	0	-2
0	0	1	0	-2

(b.1)

-2	-2	-2	1	1
-2	-2	0	1	0
-2	0	1	1	0
-2	0	1	-2	-2
-2	0	1	-2	-2

(b.2)

-2	-2	-2	1	1
-2	-2	1	1	0
-2	0	1	0	-2
0	1	1	-2	-2
1	1	-2	-2	-2

(b.3)

-2	-2	-2	-2	-2
-2	-2	0	0	0
-2	0	1	1	1
-2	0	1	-2	-2
-2	0	1	-2	-2

(b.4)

Figure 4. Line Features to clasify edges. Column (a) are the structural arrangement of pixels and (b) correspond to their matrix representations.

To determine if a pixel is within a line or a black area, a set of 5×5 line features is proposed to evaluate every pixel according to the configuration of its neighborhood. The structural representations of the line features are shown in row (a) of the Fig. 4. Notice that they model the behavior of smooth curves at pixel resolution. The first feature corresponds to straight lines (allowing a slight deviation); the second models lines with a partial tilt from the straight line; the third matches diagonals lines; and the fourth correspond to corners. The complete set of features is composed by all affine transformations of these 4 basic features. The black cells represent the approximate configuration of the edge pixels in order to be part of one-pixel-width curves. Crossed cells indicate the locations were there should not be any edge pixels and blank cell allow some variations of the basic structures.

To evaluate the features for a given pixel, first transform the images with morphological closed edges into a binary image with 1 on the edge pixels and 0 elsewhere. Then, consider the numerical representation of the features showed in Fig. 4b, as matrices and consider all its affine transformations. For a pixel to be classified as a salient edge it must score at least 5 in one of the complete set of features. If a feature is represented as

$$feature_k = \begin{bmatrix} a_{1\,1} & a_{1\,2} & a_{1\,3} & a_{1\,4} & a_{1\,5} \\ a_{2\,1} & a_{2\,2} & a_{2\,3} & a_{2\,4} & a_{2\,5} \\ a_{3\,1} & a_{3\,2} & a_{3\,3} & a_{3\,4} & a_{3\,5} \\ a_{4\,1} & a_{4\,2} & a_{4\,3} & a_{4\,4} & a_{4\,5} \\ a_{5\,1} & a_{5\,2} & a_{5\,3} & a_{5\,4} & a_{5\,5} \end{bmatrix} \quad (3)$$

then, for a pixel $p = (x, y)$, the feature is evaluated with the following formulae

$$feature_k(p) = \sum_{i=1}^{5}\sum_{j=1}^{5} a_{i\,j} \cdot (x - 3 + i, y - 3 + j) \quad (4)$$

In this way, the amount of pixels that match with the line features, denoted by ℓ, and the amount of pixels that correspond to black areas, denoted by β, can be quantified to compute the fitness value. Optimal salient extraction must have the larger amount of line edges while having the least amount of black edges. Following this, the fitness function is defined as

$$fitness(I) = \ell - \beta \quad (5)$$

where I is the image being evaluated. The value β can also be written as $\beta = Total\ edges - \ell$. Some samples of results obtained with the evolutionary approach of salient edge detection are shown in Fig. 5.

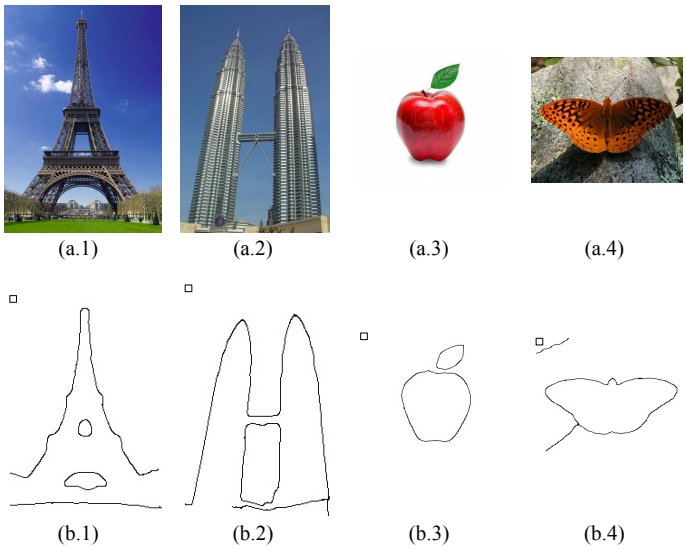


Figure 5. Some sample results obtained by the evolutionary optimization for salient edge estraction

V. Feature Description and Similarity Measure

1) Orientation Map

The resulting salient edges images will be used to form the database of the sketch-based CBIR system. The salient edge image will be transformed into an orientation map. For every edge pixel the orientation will be approximated into one of the 8 possible orientations shown in Fig. 6. The orientation map will be composed by a matrix of the same size of the image with values from 0 to 8.

The line features are useful to determine the orientation of the edge. For instance, features (a.1), (a.2) and (a.3) form Fig. 4 can be used to classify orientations 5, 4 and 3 from Fig. 6, respectively. With the complete set of features, all the orientations can be classified. For example, if an edge scores a value of at least 5 while evaluating with feature (a.2), then the label 4 will be assigned to that coordinate on the orientation map.

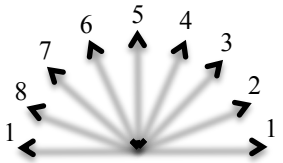


Figure 6. Eight possible orientations can be estimated to form the orientation map. Each number, n, work as a label for the corresponding angle of $\pi(1-n)/8$ radians

2) Preparing the Sketch

To begin the search process, the user must draw a sketch as a query. The sketch will be passed through a thinning algorithm to assure that it is composed of thin lines. Then, it will be transformed into an orientation map. The resulting orientation map is used to create 8 binary sub-images, where each sub-image corresponds to an orientation. For convenience, the sub-images will have passible values 0 or 255.

For each sub-image, a hit-map will be created applying distance transform that stops at d_{max}, as shown in Fig. 7. The distance transform will assign values from 0 to d_{max} to the pixels according the distance of the closest 0-valued pixel on the binary sub-image. These hit-maps will be used to match with the salient edges orientation maps on the database.

3) Similarity measure

The similarity will be a real value that can be increased iteratively according to the following pseudo code:

```
For every orientation map M on the database
   Sm=0
   For every sketch's hit-map Ho
      For every coordinate (x,y)
         If M(x,y) ≤ 8 & Ho(x,y) ≤ dmax
            If |M(x,y) − o| ≤ 1
               Sm = Sm + d
```

Where the hit map H_o is the distance-transformed sub-image of the sketch corresponding the orientation O. The value d_{max} is set to be 5. Note that the similarity measure is incremented not only when the value on the orientation map is equal to the index of the hit-map, but also with the adjacent orientations. The system will display the images whit the highest similarity measure for its orientation map.

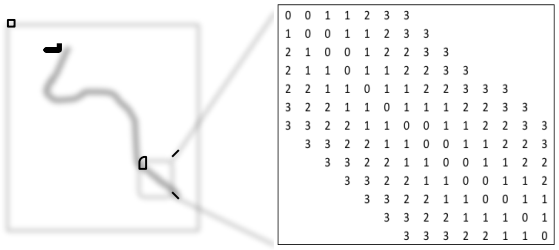


Figure 7. Zoomed distance transformed hitmap for d_{max}=3

VI. Implementation and Results

The system was implemented for a set of 100 trial images, collected from the Web. The trial image sets contains approximately 5 images for 20 categories. For example, the collection contains 5 images of the Eiffel tower, 5 images if the Petronas towers, etc., but the database is not labeled.

Some functions from the library OpenCV were used to handle the trial images. All of them were reduced to a scale of at most 256 pixels on the largest side. The salient edges were extracted for each image using the evolutionary to determine the set of parameters used on the image-processing algorithm described in section VI.

Some examples of images retrieved by a sketch are shown in Fig. 8. The resulting set corresponds to the top 6 images ranked by the similarity measure. Notice how in Fig 8a the best to matches are very similar regarding the object that is represented on it, the Eiffel tower, in almost identical position and scale across the image; but these images are presented in different color schemes and contrast, yet the feature extraction is robust under these variations. The sketched query can be presented as a general shape like in Fig. 8b or as a more complex set of lines, as shown in Fig. 8c.

VII. Conclusions and Future Work

An architecture for a queried-by-sketch image retrieval system for the Web was proposed. Two essential modules were distinguished: salient edge extraction to establish a database and edge description and matching. Salient edge detection is addressed by an evolutionary algorithm to guarantee that an optimal solution will be searched for each image. In this way, the success of the salient edge detection is not conditioned by the characteristics of the image. This method proved to be effective for a collection of diverse images.

To generate the descriptors we proposed a novel way to estimate the orientation of edges without using the gradient. The descriptors are used to compare the resulting salient edges with a sketch. Image search results of the implementation in low scale show that the proposed methodology is a viable way to search using a sketch query.

Some improves can be make in the similarity measure as the orientation an location are only local characteristics. A next step is to create refinement methods according to structural characteristics of the curves and length of the matching lines.

This will address result in better similarity measuring and partial matches will be ranked lower.

In future implementations, the amount of images will be increased to prove scalability of the method as a requirement to develop a search engine for the Web. Increasing the database will also result in a greater probability to obtain more similar images in the set of results.

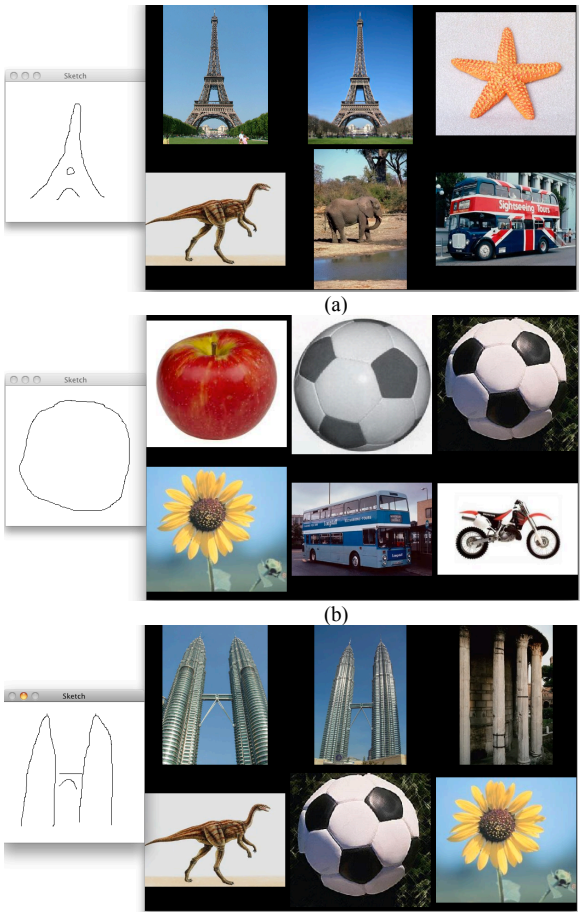


Figure 8. Retrieved images with a sketched query. Results are ordered by similarity from left to right and top to bottom. In all cases, at least a similar image is found in the set of retrieved images.

REFERENCES

[1] Eitz, Mathias, Hays, James and Alexa, Marc. "How do humans sketch objects?." *ACM Trans. Graph.* 31 , no. 4 (2012): 44.

[2] Smeulders, Arnold W. M., Worring, Marcel, Santini, Simone, Gupta, Amarnath and Jain, Ramesh. "Content-Based Image Retrieval at the End of the Early Years.." *IEEE Trans. Pattern Anal. Mach. Intell.* 22 , no. 12 (2000): 1349-1380.

[3] Datta, Ritendra, Joshi, Dhiraj, Li, Jia and Wang, James Ze. "Image retrieval: Ideas, influences, and trends of the new age.." *ACM Comput. Surv.* 40 , no. 2 (2008).

[4] "Google Image Search,", accessed October 21, 2013, http://www.google.com/imghp

[5] "TinEye Reverse Image Search" accessed October 21, 2013, http://www.tineye.com/

[6] "revImg Image Search" accessed October 21, 2013, http://www.revimg.net/

[7] "WeSee Image Search" accessed October 21, 2013, http://wesee.com/

[8] "Retrievr- Search by Sketch" accessed October 21, 2013, http://labs.systemone.at/retrievr/

[9] Grigorescu, Cosmin, Petkov, Nicolai and Westenberg, Michel A.. "Contour and boundary detection improved by surround suppression of texture edges.." *Image Vision Comput.* 22 , no. 8 (2004): 609-622.

[10] Martin, David R., Fowlkes, Charless and Malik, Jitendra. "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues.." *IEEE Trans. Pattern Anal. Mach. Intell.* 26 , no. 5 (2004): 530-549.

[11] Papari, G.; Petkov, N., "Adaptive Pseudo Dilation for Gestalt Edge Grouping and Contour Detection," *Image Processing, IEEE Transactions on* , vol.17, no.10, pp.1950,1962, Oct. 2008

[12] Hongzhi Wang; Oliensis, J., "Salient Contour Detection using a Global Contour Discontinuity Measurement," *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on* , vol., no., pp.190,190, 17-22 June 2006

[13] Songhe Feng, De Xu, Xu Yang, Attention-driven salient edge(s) and region(s) extraction with application to CBIR, Signal Processing, Volume 90, Issue 1, January 2010, Pages 1-15,

[14] Papari, Giuseppe and Petkov, Nicolai. "Edge and line oriented contour detection: State of the art.." *Image Vision Comput.* 29 , no. 2-3 (2011): 79-103.

[15] Eitz, Mathias, Hildebrand, Kristian, Boubekeur, Tamy and Alexa, Marc. "Sketch-Based Image Retrieval: Benchmark and Bag-of-Features Descriptors.." *IEEE Trans. Vis. Comput. Graph.* 17 , no. 11 (2011): 1624-1636.

[16] Belongie, Serge, Malik, Jitendra and Puzicha, Jan. "Shape Matching and Object Recognition Using Shape Contexts.." *IEEE Trans. Pattern Anal. Mach. Intell.* 24 , no. 4 (2002): 509-522.

[17] Dalal, N. and Triggs, B.. " Histograms of Oriented Gradients for Human Detection." Paper presented at the meeting of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, USA, 2005.

[18] Borgefors, Gunilla. "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm.." *IEEE Trans. Pattern Anal. Mach. Intell.* 10 , no. 6 (1988): 849-865.

[19] Cao, Yang, Wang, Changhu, Zhang, Liqing and 0001, Lei Zhang. "Edgel index for large-scale sketch-based image search.." Paper presented at the meeting of the CVPR, 2011.

[20] Canny, J.. "A computational approach to edge detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 , no. 6 (1986): 679--698.

[21] Tomasi, Carlo and Manduchi, Roberto. "Bilateral Filtering for Gray and Color Images.." Paper presented at the meeting of the ICCV, 1998.

[22] Comaniciu, Dorin and Meer, Peter. "Mean Shift Analysis and Applications.." Paper presented at the meeting of the ICCV, 1999.

[23] Storn, Rainer and Price, Kenneth. "Differential evolution--a simple and efficient heuristic for global optimization over continuous spaces." *Journal of global optimization* 11 , no. 4 (1997): 341--359.